Priority is given to read controller (4)

L47: Entry 6 of 20 File: USPT Nov 28, 2000

DOCUMENT-IDENTIFIER: US 6154796 A
** See image for Certificate of Correction **
TITLE: Apparatus and method in a network interface device for storing receiving frame status in a holding register

Abstract Text (1):
A network interface device includes a random access memory used as a transmit and receive buffer for transmission and reception of data frames between a host computer bus and a packet switched network. The network interface device includes read and write controllers for each of the transmit and receive buffers, where each write controller operates in a clock domain separate from the corresponding read controller. The read and write controllers output status information corresponding to the reading or writing of a stored data frame in the receive buffer. The memory management unit includes a synchronization circuit, which arbitrates updates to the holding registers by the read and write controllers based on the asynchronously determined presence of at least one stored data frame.

Brief Summary Text (14):
There is also a need for an arrangement that enables storage of status information in a location (i.e., holding register) separate from random access memory, enabling a read controller to immediately determine whether to transfer the stored data frame or discard the stored data frame based on reading the corresponding status information stored in the holding register to optimize efficient use of network interface resources.

Brief Summary Text (17):
Another aspect of the invention provides a network interface device for storing status information associated with a data frame, comprising a random access buffer memory, a write controller configured for writing the data frame into the random access buffer memory according to a first clock and outputting status information of said frame, a read controller configured for reading the stored data frame from the random access buffer memory according to a second clock independent from the first clock and outputting status information corresponding to the stored data frame, and a synchronization circuit asynchronously determining a presence of at least one stored data frame in the random access buffer memory, the synchronization circuit comprising a holding register and a selection circuit, the selection circuit configured for selectively storing into the holding register one of the status information output from the read controller and the status information output from the write controller. Use of the selection circuit and the synchronization circuit to selectively store the status information from the read controller or write controller in the holding register enables the synchronization circuit to reconcile potential arbitration issues between the controllers operating in their respective clock domains by granting priority to the write controller to store the corresponding status information if a full data frame is not stored in the random access memory, while granting priority to the read controller if at least one frame is stored in the random access memory. By storing the status information in a holding register, separate from the random access memory, the read controller may quickly examine whether the data frame should be read or discarded. Hence, the present invention provides the advantages of random access memory over FIFO-based buffers, including faster access speeds, smaller memory sizes and faster

flushing of non-relevant data, without the introduction of synchronization or memory contention problems.

Detailed Description Text (8):
The network interface device 10 also includes a buffer management unit 24 configured for managing DMA transfers via the DMA interface 16b. The buffer management unit 24 manages DMA transfers based on DMA descriptors in host memory that specify start address, length, etc. The buffer management unit 24 initiates a DMA read from system memory into the transmit buffer 18b by issuing an instruction to the DMA interface 16b, which translates the instructions into PCI bus cycles. Hence, the buffer management unit 24 contains descriptor management for DMA transfers, as well as pointers associated with storing and reading data from the memory portion 18. Although the buffer management unit 24 and the memory management unit 22 are shown as discrete components, the two units may be integrated to form a memory management unit controlling all transfers of data to and from the memory unit 18.

Detailed Description Text (20):
The presence of two separate clock domains 56a and 56b in writing and reading to a random access memory 18 requires that the write controller and read controller devices be coordinated and synchronized to ensure that no contention issues arise due to the relative independence of the two clock domains 56a and 56b. The SRAM MMU 22 includes a synchronization circuit 60 that asynchronously monitors the status of the RX.sub.-- SRAM 18a and 18b, enabling the memory controllers to read and write to the memory 18 between the two clock domains 56a and 56b. Thus, problems that would ordinarily arise between the two clock domains in the individual memory management units 22a, 22b, 22c and 22d are avoided by use of the synchronization circuit 60 according to a prescribed arbitration logic.

Detailed Description Text (22):
As described below, each data unit 64 is written by first reserving the contiguous buffer memory locations for the double word Status 0, Status 1 (together forming the status information 70), and the frame track 66 by writing, for example, null values into the first three contiguous memory locations, as shown in FIG. 3B. The memory controller (RM.sub.-- MMU) 22d then writes the frame data in 32-bit wide double word format into the contiguous buffer memory locations 68 following the corresponding locations for the status information 70 and the frame track field 66. The memory controller 22d continues to write the 32-bit wide frame data fields in the contiguous locations D0, D1, etc. until reception of an end of frame (RM.sub.-- ENF) signal from the MAC 20. In response to the end of frame (RM.sub.-- ENF) signal, the memory controller (RM.sub.-- MMU) 22d jumps to the first buffer memory location (Status 0--i.e., STATUS Upper), receives the status information from the media access controller, and writes the first and second portions of the status information 70 into the first and second buffer memory locations (status 0 and status 1 (i.e., STATUS Lower)), respectively. The memory controller 22d then updates the frame track field 66, then jumps to reserve the next contiguous memory locations for the status information 70 and frame track field 66 for the next received data frame (e.g., 64.sub.2).

Detailed Description Text (27):
According to the disclosed embodiment, the synchronization circuit 60 includes a read counter and a write counter for each transmit SRAM (TX.sub.-- SRAM) 18b and receive SRAM (RX.sub.-- SRAM) 18a, where each counter is configured for counting a number of written (or read) frames by changing a single bit of a counter value in response to a corresponding signal from the associated MMU controller.

Detailed Description Text (28):
FIG. 4 is a block diagram illustrating operation of the synchronization circuit 60 with the read and write controllers for the RX.sub.-- SRAM 18a according to an embodiment of the present invention. As shown in FIG. 4, the write controller

(RM.sub.-- MMU) 22d for the RX.sub.-- SRAM 18a is configured for writing a frame (e.g., the data frame 64) into the RX.sub.-- SRAM 18a according to a receive MAC clock (RMCLK) 74 synchronized relative to the network clock domain 56b. The write controller 22d, upon writing an entire data frame 64 into the RX.sub.-- SRAM 18a, outputs a write signal to the synchronization circuit 60 indicating the completed writing of the data frame 64. Specifically, the write controller 22d writes the data frame 64 in FIG. 3A by receiving the frame data 68 from the MAC 20 according to the RMCLK 74. The write controller 22d then reserves a portion of the prescribed memory location 64 of the RX.sub.-- SRAM 18a by first writing null data for the frame track 66 to hold header information. Further, the write controller, in one embodiment, may also write null data for the status information 70, reserving memory locations contiguous to the location that stores the frame track. The actual frame data 68 is then written (location "Y"), followed by status information 70 (at location "Z"). Following the writing of the status information 70 at location "Z", the write controller 22d then returns to the frame track field 66 at location "X" and updates the frame track with the appropriate header information, including setting the end of frame address (ENF ADDR), updating the count (CNT) field indicating the number of DWORDS in the frame, the frame bit (FRM), and the ENF bit indicating that the memory location 64 stores valid data.

Detailed Description Text (30):
The write controller 22d outputs the write signal to the synchronization circuit 60 after updating the frame track field 66, at which point the SRAM 18a stores a valid data frame 64 at a prescribed location. Successive writing of frames results in the RM.sub.-- MMU 22d outputting a plurality of respective status values (RX.sub.-- STATUS lower and RX.sub.-- STATUS upper) to the holding registers 84a, 84b. Assuming, however, that the read controller 22c attempted to read the first frame F1 from the memory location 64 prior to completion by the write controller 22d, the read controller 22c would obtain invalid data, since the status information field 70 would not yet be updated with valid data. In addition, the read controller 22c, upon accessing the status information 70 corresponding to a stored frame 64, supplies the read status information from the accessed memory location to the synchronization circuit 60 in order to indicate the read status. Hence, invalid status information could be written to the holding registers 84a, 84b if the read controller 22a attempted a read operation before completion of the corresponding write operation.

Detailed Description Text (31):
According to the disclosed embodiment, the synchronization circuit 60 determines a presence of a complete stored data frame 64 in the random access memory 18a in response to read and write signals and independent of the bus clock 72 and the MAC clock (RMCLK) 74. Specifically, the synchronization circuit 60 includes a write frame counter 76 configured for changing a single bit of a write counter value in response to the write signal output from the write controller 22d. The synchronization circuit 60 also includes a read frame counter 78 configured for changing a single bit of a read counter value in response to the read signal from the read controller 22c. As described above, the read controller 22c is configured for reading the frame 64 from the receive SRAM 18a according to a host bus clock (BCLK) 72, where the read controller 22c outputs a read signal to the synchronization circuit 60 in response to the reading of the frame.

Detailed Description Text (37):
The use of a status holding register 84, that is resident in the synchronization circuit 60, provides a more efficient read operation by the read controller RB.sub.-- MMU 22c. During a write operation by RM.sub.-- MMU 22d to the RX.sub.-- SRAM 18a, the STATUS Upper and STATUS Lower fields 70, denoted Status 0 and Status 1 respectively in the FIG. 4, are concurrently written to the holding registers 84a, 84b. These holding registers 84a and 84b are also updated by the RB.sub.-- MMU 72 when it performs a read of the RX.sub.-- SRAM 18a. As the read controller 22c retrieves data frames, it writes the status information data into the holding

registers 84a, 84b. Consequently, the receive frame status can be immediately accessed and updated without needing to manipulate the receive buffer pointer back and forth. Hence, overall latency of the network interface device is reduced.

Detailed Description Text (41):
Storage of the status information in the holding registers 84a, 84b enables a read controller (such as the RB.sub.-- MMU 22c) or a host performing a slave transfer, to access the received frame status before actual retrieval of the received frame data. In addition, rapid and efficient data transfers can be performed from the SRAM 18a by a host operating the network interface device in slave mode, since determination of whether to read a stored data frame or "flush" the frame can be quickly determined by reading the status and the frame track fields, and then merely incrementing a read pointer to begin retrieval of the stored frame data 68.

CLAIMS:

17. The network interface device of claim 11, wherein the frame comprises tracking information, the valid bit being derived from the tracking information, and the start processing bit being derived from signals outputted from the read controller, the write controller, and the synchronization circuit.